Homework # 5

1. Read Lecture 10 of Trefethon and Bau (see the Course Documents-Books folder) on Householder reflections.
   (a) Let $\|v\| = 1$, show that $I - 2vv^T$ is an orthogonal matrix.
   (b) Let A be an $n \times p$ matrix with $n > p$ and assume that A is full rank (i.e. the columns are linearly independent). Derive the complexity of using Householder reflection to from the QR decomposition of A. (Trefethon and Bau do this in the Lecture, but try first to do it on your own.)
   (c) Code a function `QR_householder(A)` that forms the QR decomposition of the matrix A using Householder reflection. Besides standard python, your function should only make use of numpy's matrix-vector arithmetic. Show some examples demonstrating that your function is correct.
2. In Lecture 4 of Trefethon and Bau, do exercise 4.1.
3. In Lecture 7 of Trefethon and Bau, do the following exercises.
   - 7.2
   - 7.3 (recall that det(AB)=det(A)det(B) and use the QR decomposition)
   - 7.4 (the $P^{(1)}$ and $P^{(2)}$ of the problem are the spans of the given vectors)
   - 7.5 (Note that you cannot assume that the QR is formed by Gramm-Schmidt.)

   The QR we discussed in class is Trefethon's reduced rank QR. For the reduced rank QR, if $X = QR$ then X and Q have the same dimensions. Trefethon distinguishes the full rank QR in which Q is extended to be square.
4. Attached you will find a file `X.mtx`. The mtx file format is used to store sparse matrices. Use scipy's sparse library to load and manipulate the sparse matrix. For some orientation, see the attached `sparse.py`.

(a) Use a power iteration to compute the first two dominant eigenvalues/vectors of XX^T. Use only matrix-vector arithmetic. You can compare to the correct values as computed by `scipy.sparse.linalg.eigsh(mm, k=2)`. `eigsh` uses a Lanczos iteration which we will discuss next week.

(b) You can convert X to a dense matrix by

```
Xdense = X.toarray()
```

`Xdense` is a numpy matrix. Try to repeat (a) but using `Xdense`.